Received: 10 March 2023 | Revised: 4 April 2023 | Accepted: 15 April 2023 | Published online: 21 April 2023

RESEARCH ARTICLE

Comparing BERT against Traditional Machine Learning Models in Text Classification

Journal of Computational and Cognitive Engineering
yyyy, Vol. XX(XX) 1–5
DOI: 10.47852/bonviewJCCE3202838

BON VIEW PUBLISHING

Eduardo C. Garrido-Merchán[1,*], Roberto Gozalo-Brizuela[1], Santiago González-Carvajal[2]

1 Quantitative Methods Department, Universidad Pontificia Comillas, Spain
2 Artificial Intelligence Department, Universidad Politécnica de Madrid, Spain

*Corresponding author: Eduardo C. Garrido-Merchán: *Quantitative Methods Department, Universidad Pontificia Comillas, Spain, ecgarrido@comillas.edu.*

Abstract: The BERT model has arisen as a popular state-of-the-art model in recent years. It is able to cope with NLP tasks such as supervised text classification without human supervision. Its flexibility to cope with any corpus delivering great results has make this approach very popular in academia and industry. Although, other approaches have been used before successfully. We first present BERT and a review on classical NLP approaches. Then, we empirically test with a suite of different scenarios the behaviour of BERT against traditional TF-IDF vocabulary fed to machine learning models. The purpose of this work is adding empirical evidence to support the use of BERT as a default on NLP tasks. Experiments show the superiority of BERT and its independence of features of the NLP problem such as the language of the text adding empirical evidence to use BERT as a default technique in NLP problems.

Keywords: BERT, natural language processing, machine learning, comparison

1. Introduction

Natural Language Processing (NLP) methodologies have flourished and lots of papers solving different tasks of the field, such as text classification (Aggarwal & Zhai, 2012), named entity recognition (Nadeau, 2007) or summarization (Puente et al., 2013), have been published. We can differentiate, mainly, between two types of approaches to NLP problems: Firstly, linguistic approaches (Cambria & White , 2014) that generally use different features of the text that the experts on the domain consider that are relevant have been extensively used. Those features could be combinations of words, or n-grams (Stamatatos, 2011), grammatical categories, unambiguous meanings of words, words appearing in a particular position, categories of words and much more. These features could be built manually for an specific problem or can be retrieved by using different linguistic resources (Besanccon et al., 2010) such as ontologies (Busch et al., 2006).

On the other hand, Machine Learning (ML) (Manning & Schutze, 1999) and deep learning based approaches (Otter et al., 2020) that classically have analyzed annotated corpora of texts inferring which features of the text, typically in a bag of words fashion (Zhang et al., 2010) or by n-grams, are relevant for the classification automatically. Both approaches have their pros and cons, concretely, linguistic approaches have great precision but their recall is low as the context where the features are useful is not as big as the one processed by machine learning models. Although, the precision of classical NLP systems was, until recently, generally better as the one delivered by machine learning (Garrido & Lera , 2015). Nevertheless, recently, thanks to the rise of computation, machine learning text classification dominates in scenarios where huge sizes of texts are processed.

Generally, linguistic approaches consist in applying a series of rules, which are designed by linguistic experts (Khurana et al., 2023). An example of linguistic approach can be found at (Hutto & Gilbert, 2014). The advantage of these type of approaches over ML based approaches is that they do not need large amounts of data. Regarding ML based

approaches, they usually have a statistical base (Khurana, 2017). We can find many examples of these type of approaches: BERT (Devlin et al., 2018), Transformers (Vaswani et al., 2017) etc.

Another issue with traditional NLP approaches is multilingualism (Bikel & Zitouni, 2012). We can design rules for a given language, but sentence structure, and even the alphabet, may change from one language to another, resulting in the need to design new rules. Some approaches such as the Universal Networking Language (UNL) standard (Uchida & Zhu, 2001) try to circumvent this issue, but the multilingual resource is hard to build and requires experts on the platform. Another problem with UNL approaches and related ones, would be that, given a specific language, the different forms of expression, i.e. the way we write in, for example, Twitter, is very different from the way we write a more formal document, such as a research paper (Farzinder & Inkpen, 2015).

Bidirectional Encoder Representations from Transformers (BERT) is a NLP model that was designed to pretrain deep bidirectional representations from unlabeled text and, after that, be fine-tuned using labeled text for different NLP tasks (Devlin et al., 2018). That way, with BERT model, we can create state-of-the-art models for many different NLP tasks (Devlin et al., 2018). We can see the results obtained by BERT in different NLP tasks at Devlin et al. (2018).

In this work we compare BERT model (Devlin et al., 2018) with a traditional machine learning NLP approach that trains machine learning models in features retrieved by the Term Frequency - Inverse Document Frequency (TF-IDF) (Zhang et al., 2005) algorithm as a representative of these traditional approaches (Trstenjak et al., 2014). With this technique, we avoid the construction of a linguistic resource that need expert supervision, simulating it with the punctuation retrieved for any term by the TF-IDF technique. We lose precision by doing this operation but gain recall.

We have carried out four different experiments about text classification. In all of them, we have used two different classifiers: BERT and a traditional classifier created in the way that we have just explained. In this work we start by presenting some related work, then, we describe the models we have used in our experiments, after that, we describe the experiments we have carried out and show the obtained results and, finally, we present the conclusions drawn from the work and some future lines of work.

2. Literature Review

In this section, we summarize the main comparisons against advanced models such as the BERT transformer and classical natural language processing. Recently, BERT has achieved state-of-the-art results in a broad range of NLP tasks (Devlin et al., 2018), so the question that is discussed is whether classical NLP techniques are still useful in comparison to the outstanding behaviour of BERT and related models.

It is interesting to study how does the BERT model represent the steps of the traditional NLP pipeline (Tenney et al., 2019) in order to make a fair comparison. The main conclusion of this paper is that their work shows that the model adapts to the classical NLP pipeline dynamically, revising lower-level decisions on the basis of disambiguating information from higher-level representations. In other words, we can think of BERT as a generalization of the traditional NLP pipeline, hence being more dynamic.

An argument that defends classical machine learning NLP approaches is that the BERT approach need huge amounts of texts to deliver proper results. An interesting work, Usherwood and Smit (2019) that focus on a pure empirical comparison of BERT and ULMFiT (Rother & Rettberg, 2018) w.r.t traditional NLP approaches in low-shot classification tasks where we only have 100-1000 labelled examples per class shows how BERT, representing the best of deep transfer learning, is the best performing approach, outperforming top classical machine learning models thanks to the use of transfer learning (Devlin et al., 2018). In our work, we are going to test this hypothesis under different problems that also involve texts in different languages.

A common critique of classical NLP practitioners is that the BERT model and machine learning methodologies can be fooled easily, commiting errors that may be severe in certain applications and that can be easily solved by symbolic approaches. Following this reasoning, in this work (Jin et al., 2019) the authors present the TextFooler baseline, that generates adversarial text in order to fool BERT's classification (Jin et al., 2019). We wonder if these experiments are representative of common scenarios and hypothesize that, although it is true that some texts may fool BERT, they are not representatives of common problems. In order to test this hypothesis, we are going to measure the results given by BERT in different languages. If BERT fail in these problems, then these adversaries may be common. Although, if BERT outperforms classical approaches under standard circumstances, then we can state that these adversarial attacks may not be common.

3. The BERT model and the traditional machine learning NLP methodology

Having reviewed related work, we will now introduce the traditional NLP approaches that we are comparing with BERT and then, the details of the BERT model.

3.1. Term Frequency - Inverse Document Frequency (TF-IDF)

A classical way to deal with a supervised learning NLP task is to build a bag of-words model with the most weighted words given by the TF-IDF algorithm.

Assuming there are N documents in the collection, and that term t_i occurs in n_i of these documents. Then, inverse document frequency can be computed as:

$$idf(t_i) = log\frac{N}{n_{i.}} \quad (1)$$

Actually, the original measure was an integer approximation to this formula, and the logarithm was base 2. However, Aggarwal and Zhai (2012) are the most commonly cited form of IDF. For more information we refer the reader to the original source (Robertson, 2004).

On the other hand, given a term t_i, we denote by tf_i the frequency of the term t_i in the document under consideration (Robertson, 2004).

Finally, TF-IDF is defined for a given term t_i in a given document as follows:

$tfidf(t_i) = tf_i \cdot idf(t_i)$.

In our experiments, regarding the standard NLP algorithms, we will be using TF-IDF to build a vocabulary for a machine learning model. Further details are introduced in the experiments section.

3.2 Bidirectional Encoder Representations from Transformers (BERT)

We now explain what we consider to be the state-of-the-art technique on natural language processing. Regarding the BERT model, there are two steps in its framework: *pre-training* and *fine-tuning* (Devlin et al., 2018). During pre-training, the model is trained on unlabeled large corpus. For fine-tuning, the model is initialized with the pre-trained parameters, and all the parameters are fine-tuned using labeled data for specific tasks.

BERT's model architecture is a multi-layer bidirectional Transformer encoder (Devlin et al., 2018) based on the original implementation described in Vaswani et al. (2017).

This kind of encoder is composed of a stack of $N = 6$ identical layers. Each of these layers has two sub-layers. The first one is a multi-head self-attention mechanism, and the second one, is a simple position-wise fully connected feedforward network. It employs a residual connection (He et al., 2016) around both sub-layers, followed by a layer normalization (Ba, 2016). That is, the output of each sub-layer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is the function implemented by the sub-layer (Vaswani et al., 2017).

In relation to, multi-head self-attention, first, we need to define scaled dotproduct attention. It is define as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

,

where Q is the matrix of queries, K is the matrix of keys, V is the matrix of values and d_k is the dimension of the Q and K matrices. Now, we can define multi-head attention as

$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O$,

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$. Multi-head attention consists on projecting the queries, keys and values h times with different, learned linear projections to d_k, d_k and d_v (dimension of the values matrix), respectively. Then, on each of these projected versions of the queries, keys and values, we perform the attention function in parallel, yielding in d_v-dimensional output values. Finally, these are concatenated and projected, resulting in the final values (Vaswani et al., 2017). Self-attention means that all of the keys, values and queries come from the same place.

BERT represents a single sentence or a pair of sentences (for example, the pair *(question,answer)*)as a sequence of tokens according to the following features: BERT uses WordPiece embeddings (Wu et al., 2016). The first token of the sequence is "[CLS]". When there is a pair of sentence, in the sequence, they are separated by the "[SEP]" token. And, an embedding is added to every token indicating whether it belongs to the first or the second sentence. For a given token, its input representation is constructed by summing the corresponding token, position, and segment embeddings (Devlin et al., 2018).

Pre-training is divided into: *Masked LM* and *Next Sentence Prediction (NSP)*. The first one, consists in masking some percentage of the input tokens at random (using the "[MASK]" token), and then, predict those masked tokens. The second one consists in, given two sentences A and B, 50% of the time B is the actual next sentence that follows A (labeled as IsNext), and 50% of the time B is a random sentence from the corpus (labeled as NotNext) (Devlin et al.,2018).

Fine-tuning is straightforward since the self-attention mechanism in the Transformer allows BERT to model many downstream tasks. For each task, we simply plug in the specific inputs and outputs into BERT and fine-tune all the parameters (Devlin et al.,2018).

A random sampling method was employed for the study. The sample for the study consisted of 22 first year in-service postgraduate science teachers from one of the Colleges of Education in Bhutan. 13 male (59%) teachers and 9 female (40.9%) teachers participated in the study. The sample comprised of 7 biology teachers (31.8%), 10 chemistry teachers (45.5%), and 5 Physics teachers (22.7%). Since the participation for this study was purely on a voluntary basis, only 22 out of 39 in-service postgraduate science teachers took part in the study. The overall response rate was recorded at 56%.

4. Experiments

In order to compare BERT model with respect to the traditional machine learning NLP methodology, we have designed four experiments that are described throughout the section.

In these experiments, we will be using TfidfVectorizer from sklearn Python 3 module. After using TF-IDF to preprocess the text, we will be using Predictor from auto ml module (in the third and fourth experiments), and H2OAutoML from h2o module (in the second experiment), to find the best model to fit the data. In the first experiment, we will, instead, show how much work needs to be done in order to get close to the results obtained, with no effort, using BERT model. For this purpose, we will be using many sklearn models and study their results in depth.

Regarding BERT's implementation, we have used the pre-trained BERT model from ktrain Python 3 module. This model expects the following directory structure: a directory which must contain two subdirectories: **train** and **test**. Each one of them, in turn, must contain one subdirectory per class (named after the name of the class they represent). And, finally, each class directory, must contain the '.txt' files (their name is irrelevant) with the texts that belong to the class they represent.

4.1. IMBD Experiment

In the first experiment, we have downloaded the IMDB dataset from the following website. It contains 50000 movie reviews (25000 to train the model and 25000 to test it) to perform sentiment analysis, a popular supervised learning text classification task. The dataset is classified into two different classes: Positive and negative movie reviews.
We have compared the behaviour of a pre-trained default BERT model w.r.t different popular machine learning models such as SVC or Logistic Regression that use a vocabulary extracted from a TF-IDF model obtaining the following results:
Table 1. Accuracy retrieved by the different methodologies in the IMDB experiment over the validation set.

Model	Accuracy
BERT	**0.9387**
Voting Classifier	0.9007
Logistic Regression	0.8949
Linear SVC	0.8989
Multinomial NB	0.8771
Ridge Classifier	0.8990
Passive Aggresive Classifier	0.8931

As we can see, BERT outperforms the rest of the models. It is noteworthy that obtaining these results with the traditional approaches has been far more complicated than obtaining this result with BERT.

4.2. RealOrNot tweets experiment

Our second experiment deals with the RealOrNot tweets written in English. We have downloaded the dataset from the following website. The task to solve here is pure binary text classification. It contains tweets classified into two different classes: Tweets about a real disaster and tweets which are not about a real disaster.

We have just used the *tweet* and *class* columns. We have also used the re Python 3 module to preprocess the tweets (#anything − > hashtag, @anyone − > entity, etc.). After that, we have generated the directory structure that we need to use BERT model (using 75% data to train and 25% data to validate). The obtained results have been summarized in the following table:

Table 2. RealOrNot experiment results.

Model	Accuracy	Kaggle Score
BERT	**0.8361**	**0.83640**
H2OAutoML	0.7875	0.77607

Finally, we have classified the data from the Kaggle competition with BERT. We have scored **0.83640**. We can see this result here (Santiago Gonz´alez). Regarding the traditional approaches, the best classifier from the h2o module has turned out to be the H2OStackedEnsembleEstimator : Stacked Ensemble with model key StackedEnsemble BestOfFamily AutoML 20200221 120302. And, its score in the competition has been **0.77607**.

4.3. Portuguese news experiment

Description Having seen that BERT has outperformed an AutoML technique and other classical machine learning models using a vocabulary built from a traditional NLP technique such as TF-IDF in the English language, we choose to change the language to see if the BERT model also behaves well. We have downloaded the Portuguese news dataset from the following website. It contains articles from the news classified into nine different classes: ambiente, equilibrioesaude, sobretudo, educacao, ciencia, tec, turismo, empreendedorsocial and comida.

We have just used the *article text* and *class* columns. We have generated the directory structure that we need to use BERT model (using 75% data to train and 25% data to validate obtaining the following results:

Table 3. Portuguese news experiment results.

Model	Accuracy	Kaggle Score
BERT	**0.9093**	**0.91196**
Predictor (auto ml)	0.8480	0.85047

Finally, we have classified the data for the Kaggle competition scoring a **0.91196** accuracy. We can see this result here (Santiago González). Regarding the traditional methods, the best classifier has turned out to be a GradientBoostingClassifier. And, the score in the competition of this model has benn **0.85047**.

4.4. Chinese hotel reviews experiment

Description Our last experiment involves a completely different language, Peninsular Chinese simplified characters zh-CN, where we hypothesize that, given that the way of expressing this Language is through different symbols that are not separated by spaces BERT may not output a good result. The experiment is a sentiment analysis problem involving Chinese hotel reviews. We have downloaded the dataset from the following website. It contains hotel reviews classified into two different classes: Positive hotel reviews and negative hotel reviews.

In this experiment, we have used 85% of the data to train the model and 15% of the data to validate it. Results are given in the following table:

Table 4. Chinese hotel reviews results.

Model	Accuracy
BERT	**0.9381**
Predictor (auto ml)	0.7399

We can observe how, independently of the language and its characteristics, BERT behaviour outperforms classical NLP approach.

Finally, we have tried to do some predictions with BERT using Google Translator. For example, we have tried to predict a class for: 这家酒店的风景和服务都非常糟糕 , which means: "the view and service of this hotel are very bad". The predicted class for this hotel review has been **neg**, which is correct.

Regarding the traditional approaches, the best model has turned out to be a GradientBoostingClassifier. But in this case, the model has been pretty bad, since the probability for both classes is very close. In this experiment, the importance of transfer learning has become apparent, since the dataset was pretty small compared to the ones used in the previous experiments.

5. Conclusions and further work

In this work we have introduced the BERT model and the classical NLP strategy where a machine learning model is trained using the features retrieved with TFIDF and hypothesize about the behaviour of BERT w.r.t these techniques in the search of a default technique to tackle NLP tasks. We have introduced four different NLP scenarios where we have shown how BERT has outperformed the traditional NLP approach, adding empirical evidence of its

superiority in average NLP problems w.r.t. classical methodologies. It is also noteworthy the importance of transfer learning. We have been able to obtain this results thanks to pre-training. Transfer learning has become more apparent in experiment 4.4 (which has the smallest dataset among all the experiments). We are nevertheless aware of the limitations of the BERT model. Although it seems that it is a good default for NLP tasks, its results can be improved. In order to do so, we would like to research in a hyperparameter auto-tuned BERT model for any new NLP task with Bayesian Optimization. We would like to use that autotuned BERT to enable classification of language messages for robots (Merchan & Molina, 2020; Garrido-Merchán et al., 2020) showing consciousness correlated behaviours.

Recommendations

The finding revealed that the lack of training for both teachers and students was the main factor that prevented them from using educational technology tools in teaching and learning Ecology. Therefore, training on educational technology for both teachers and students is recommended. Since educational technology tools have arose excitement and curiosity amongst students, they recommended other module tutors to use educational technology tools as well. Educational technology tools integrated in the module will be further replicated by students teacher during teaching practice or as a full fledge teacher. Therefore tutors were recommended to use variety of educational technology tools in learning, teaching and an assessment.

Acknowledgements

The authors gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at Universidad Autónoma de Madrid. The authors also acknowledge financial support from Spanish Plan Nacional I+D+i, grants TIN2016-76406-P and TEC2016-81900-REDT.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

References

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In Mining Text Data, 163-222.

Besançon, R., De Chalendar, G., Ferret, O., Gara, F., Mesnard, O., Laïb, M., & Semmar, N. (2010, May). LIMA: A multilingual framework for linguistic analysis and linguistic resources development and evaluation. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation, 3697-3704.

Bikel, D., & Zitouni, I. (2012). Multilingual natural language processing applications: from theory to practice. USA: IBM Press.

Busch, J. E., Lin, A. D., Graydon, P. J., & Caudill, M. (2006). U.S. Patent No. 7,027,974. Washington, DC: U.S. Patent and Trademark Office.

Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. IEEE Computational Intelligence Magazine, 9(2), 48-57.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12(2011), 2493-2537.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint:1810.04805.

Farzindar, A., & Inkpen, D. (2015). Natural language processing for social media. Synthesis Lectures on Human Language Technologies, 8(2), 1-166.

Garrido-Merchán, E. C., Molina, M., & Mendoza, F. M. (2020). An artificial consciousness model and its relations with philosophy of mind. arXiv preprint:2011.14475.

Green Jr, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961). Baseball: an automatic question-answerer. In Western Joint IRE-AIEE-ACM Computer Conference, 219-224.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the International AAAI Conference on Web and Social Media, 8 (1), 216-225.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2019). Is bert really robust? natural language attack on text classification and entailment. arXiv preprint:1907.11932, 2.

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. Multimedia Tools and Applications, 82(3), 3713-3744.

Manning, C., & Schutze, H. (1999). Foundations of statistical natural language processing. USA: MIT Press.

Merchán, E. C. G., & Molina, M. (2020). A machine consciousness architecture based on deep learning and gaussian processes. In Hybrid Artificial Intelligent Systems: 15th International Conference, 15(350-361).

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. IEEE Transactions on Neural Networks and Learning Systems, 32(2), 604-624.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532-1543.

Puente, C., Olivas, J. A., Garrido, E., & Seisdedos, R. (2013). Creating a natural language summary from a compressed causal graph. In 2013 Joint ifsa World Congress and nafips Annual Meeting, 513-518.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of Documentation, 60(5), 503-520.

Rother, K., & Rettberg, A. (2018). Ulmfit at germeval-2018: A deep neural language model for the classification of hate speech in german tweets. In 14th Conference on Natural Language Processing, 113-119.

Scott, S. (1998). Feature engineering for a symbolic approach to text classification. Canada: University of Ottawa.

Stamatatos, E. (2011). Plagiarism detection using stopword n-grams. Journal of the American Society for Information Science and Technology, 62(12), 2512-2527.

Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. arXiv preprint:1905.05950.

Trstenjak, B., Mikac, S., & Donko, D. (2014). KNN with TF-IDF based framework for text categorization. Procedia Engineering, 69, 1356-1364.

Uchida, H., & Zhu, M. (2001). The universal networking language beyond machine translation. In International Symposium on Language in Cyberspace, 26-27.

Usherwood, P., & Smit, S. (2019). Low-shot classification: A comparison of classical and deep transfer machine learning approaches. arXiv preprint:1907.07543.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

Wermter, S. (1997). Hybrid approaches to neural network-based language processing. USA: International Computer Science Institute.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint:1609.08144.

Zhang, Y. Z., Gong, L., & Wang, Y. C. (2005). An improved TF-IDF approach for text classification. Journal of Zhejiang University-Science A, 6, 49-55.

Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1, 43-52.